



RECEIVED

OCT 22 2004

Technology Center 2100

Serial No. 09/837,731

Docket No. NEC OSP-10546

Replacement Figures 1, 4, 5, 8 and 11
Submitted with Amendment A



Figure 1

Modified SSA-conversion process

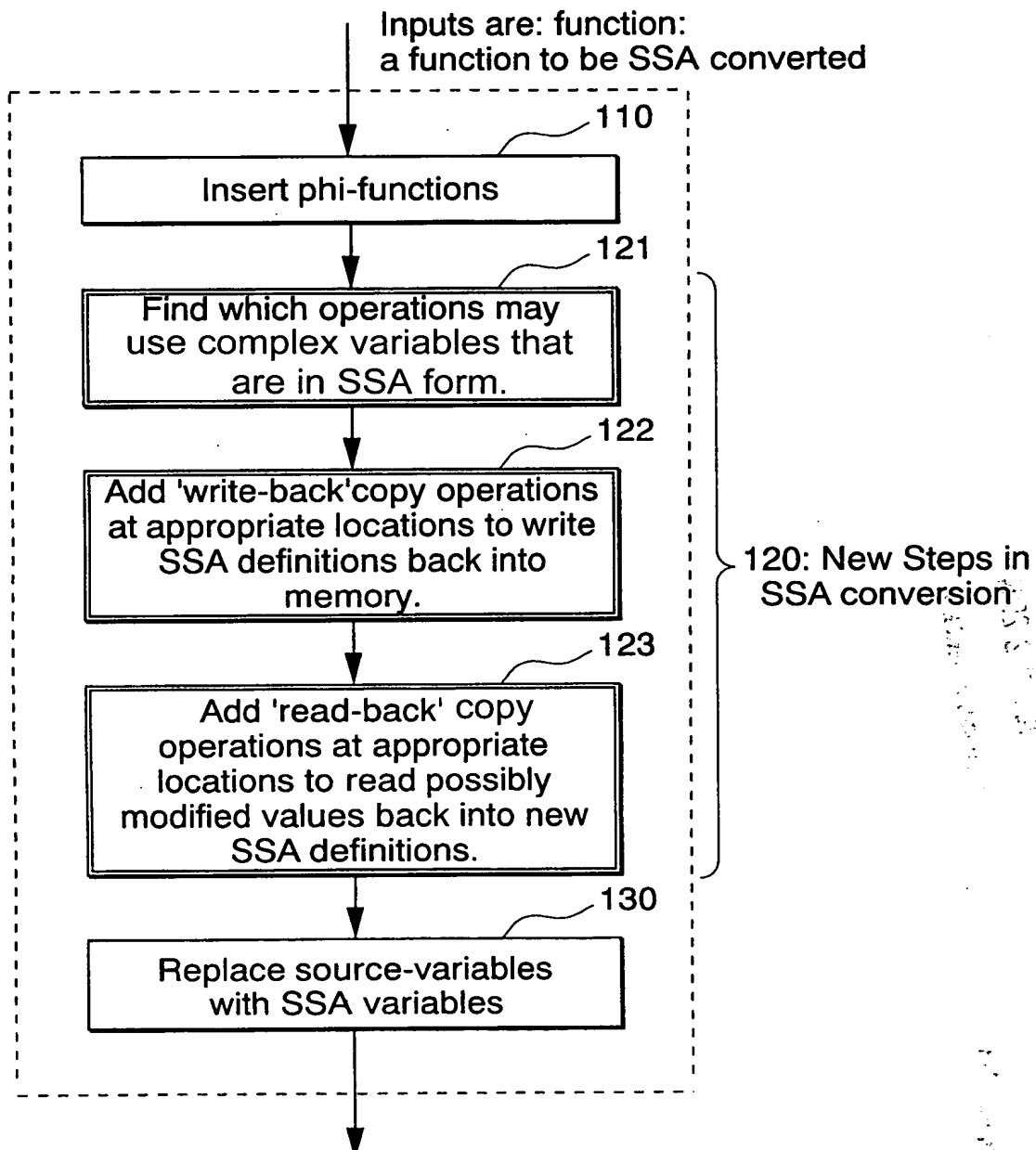
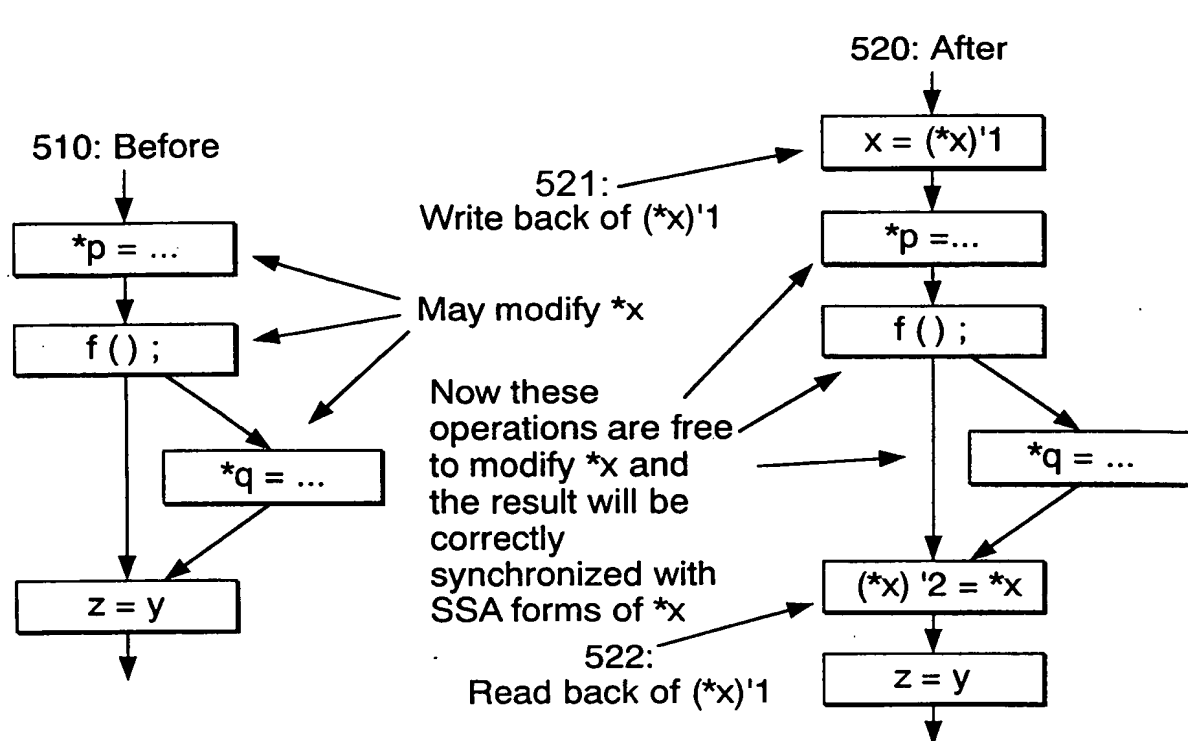




Figure 4

Placement of read/write-backs for the SSA form of $*x$, $(*x)'1$



530: A more naive method for synchronization introduces many read/write-backs

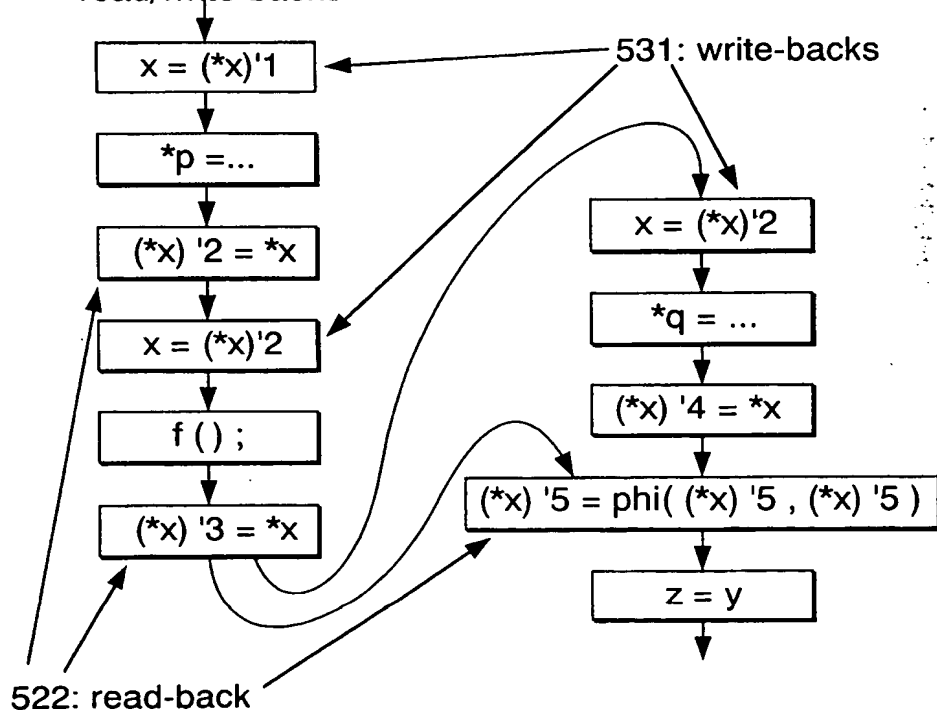




Figure 5
The procedure 'add_syncs_and_write_backs'

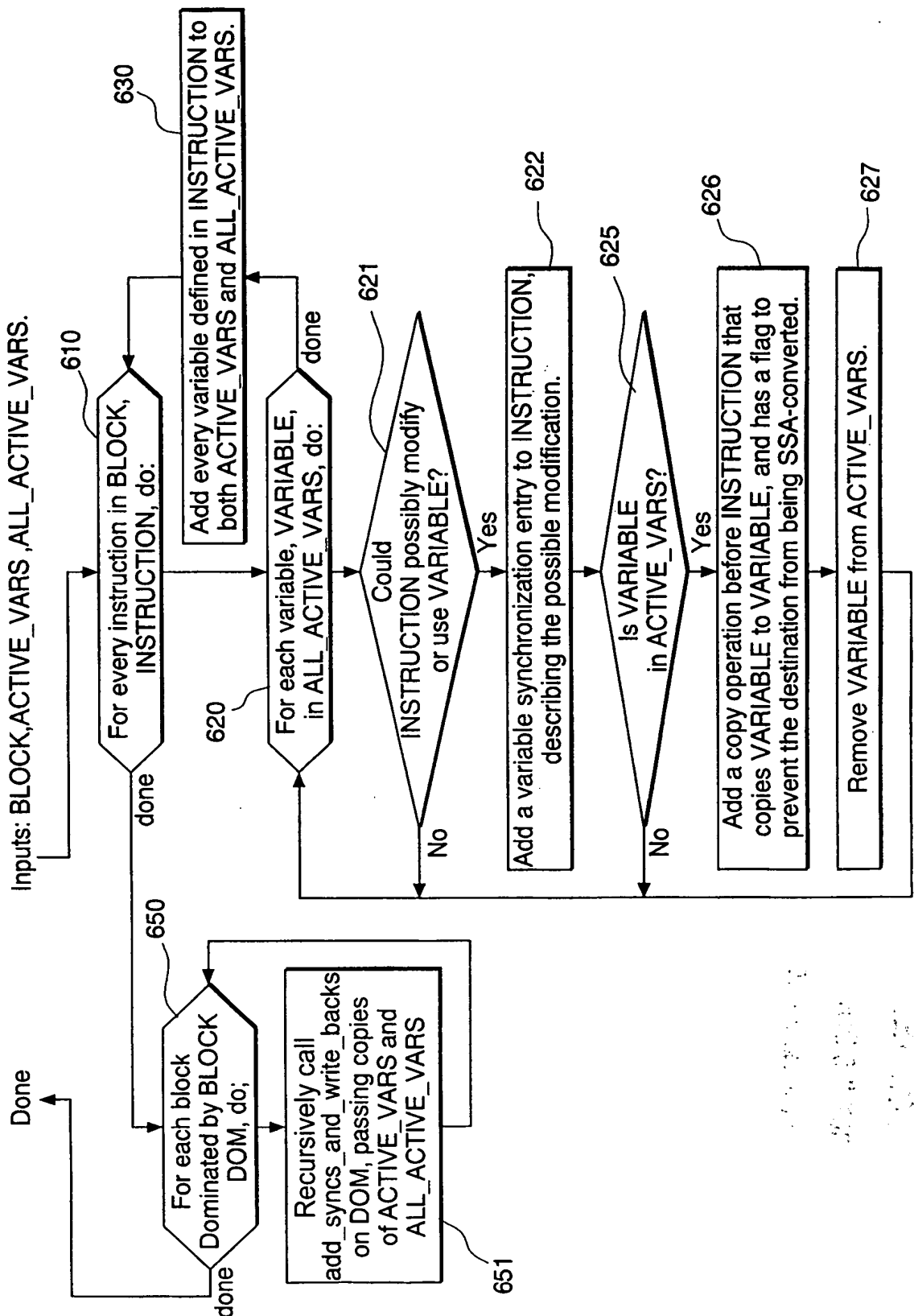




Figure 8

Example source program

This short C program is used to illustrate the invention:

```
extern int g () , h () , i () , x;  
int foo (int *p)  
{  
    (*p) ++;                                     [810]  
    if (*P > 10)  
    {  
        g () ;  
        h () ;  
        if (x > 5 )  
            g () ;  
        if (x > 3)  
            i () ;  
        else  
            X = *p;  
        *P = 5;  
    }  
    return *p;  
}
```

Here's the same program converted to a slightly more primitive form:

```
int foo (int *p)  
{  
    block1:  
        *p := *p + 1;                                     [820]  
        if (*P <= 10)  
            goto block8;  
    block2:  
        g () ;  
        h () ;  
        if (x <= 5)  
            goto block4;  
    block3:  
        g () ;  
    block4:  
        if (x > 3)  
            goto block6;  
    block5 :  
        x := *p;                                         [840]  
        goto block7;  
    block6 :  
        i () ;  
    block7 :  
        *p := 5;                                         [830]  
    block8:  
        return *p;
```



Figure 11

Register-allocated and SSA-unconverted program

using BBA-form requires having a good register allocator that will merge variables where possible, as it tends to generate a lot of variables with short lifetimes. We assume that here.

RECEIVED

OCT 22 2004

Technology Center 2100

```
int foo (int *p)
{
    int pv;

    block1:
        pv = *p + 1;
        if (pv <= 10)
            goto block8;

    block2:
        'P = pv;          /* This writes-back pv to *P. */
        g ();
        h ();
        if (x <= 5)
            goto block4;

    block3 :
        g ();

    block4 :
        if (x > 3)
            goto block6;

    block5:
        x=*p;
        goto block7;

    block6:
        i ();

    block7:
        pv =5;

    block8:
        *P= pv          /* This writes-back PV to *P. */

    return pv;
}
```